OFFICIAL

Due Date: November 3, 2003

RECEIVED
CENTRAL FAX CENTER
NOV 0 3 2003

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	
Inventor: Karthikeyan Ramasamy et al.	Examiner: Harold E. Dodds
Serial #: 09/449,085	Group Art Unit: 2177
Filed: November 24, 1999	Appeal No.: _____
Title: QUERY MONITOR PLAYBACK MECHANISM FOR POST-MORTEM PERFORMANCE ANALYSIS	

BRIEF OF APPELLANT

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 CFR §1.192, Appellants hereby submit the Appellants' Brief on Appeal from the final rejection in the above-identified application, in triplicate, as set forth in the Office Action dated June 3, 2003.

Please charge the amount of $330 to cover the required fee for filing this Appeal Brief as set forth under 37 CFR §1.17(c) to Deposit Account No. 50-1673 of the NCR Corporation, the assignee of the present application. Also, please charge any additional fees or credit any overpayments to Deposit Account No. 50-1673.

11/10/2003 AWHITE1 00000001 501673 09449085
01 FC:1402 330.00 DA

I. REAL PARTY IN INTEREST

The real party in interest is NCR Corporation, the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

III. STATUS OF CLAIMS

Claims 36-74 are pending in the application.

Claims 36, 43, 45, 49, 56, 58, 62, 69, and 71 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, II et al., U.S. Patent No. 6,230,312 (Callahan), Epperson et al., U.S. Patent No. 5,754,771 (Epperson), Posse, U.S. Patent No. 5,544,175 (Posse), and Bhargava et al., U.S. Patent No. 5,680,603 (Bhargava).

Claims 38, 51, and 64 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, and Bhargava as applied to claims 37, 50, and 63 respectively, further in view of Kimmerly et al., U.S. Parent No. 5,628,017 (Kimmerly) and Rhodes et al., U.S. Patent No. 6,073,110 (Rhodes).

Claims 39, 52, and 65 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Junkin, U.S. Parent No. 6,493,717 (Junkin) and Harel, U.S. Patent No. 5,873,081 (Harel).

Claims 40, 53, and 66 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, Bhargava, Junkin, and Harel as applied to claims 39, 52, and 65 respectively, and further in view of Zhou et al., U.S. Patent No. 5,995,511 (Zhou).

Claims 41, 42, 54, 55, 67, and 68 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, Bhargava, Junkin, Harel, and Zhou as applied to claims 40, 53, and 66, and further in view of Bamford et al., U.S. Patent No. 6,243,702 (Bamford).

Claims 44, 57, and 70 are rejected under 35 U.S.C. §1039a) as being unpatentable over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Naidu et al., U.S. Patent No. 5,752,002 (Naidu).

Claims 46, 48, 59, 61, 72, and 74 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Filepp et al., U.S. Patent No. 5,594,910 (Filepp).

Claims 47, 60, and 73 are rejected under 35 U.S.C. §103(a) as being unpatentable over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Carey et al., U.S. Patent No. 6,285,997 (Carey).

IV. STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the final Office Action.

V. SUMMARY OF THE INVENTION

Briefly, the Applicants' invention is embodied in a method of monitoring an execution of a query performed by a database system. The database system has a query coordinator (104) and at least one data server (130) as illustrated in FIG. 1, and discussed in the specification at page 4, line 7 through page 8, line 11. and the query execution comprises at least one execution thread. For each thread, the method generates first execution trace information in the query coordinator 104. Importantly, the first execution trace information comprises an execution plan in terms of one or more operator trees as shown in FIGs 3A-3C and described in the associated text. Also, for each thread, a second execution trace information is generated in the data server. Finally, the first execution trace information and the second execution trace information is written to at least one execution log file, as described in page 12, line 16 through page 14, line 24.

VI. ISSUES PRESENTED FOR REVIEW

Whether claims 38, 40, 51, 53, 64, and 66 are indefinite under 35 U.S.C. § 112 for failing to particularly point out in distinctly claim the subject matter which the Applicants regard as the invention.

Whether claims 36, 43, 45, 49, 56, 58, 62, 69, and 71 are patentable under 35 U.S.C. §103(a) Callahan, Epperson, Posse, and Bhargava.

Whether claims 38, 51, and 64 patentable under 35 U.S.C. §103(a) over Callahan, Epperson,

Posse, and Bhargava as applied to claims 37, 50, and 63 respectively, further in view of Kimmerly and Rhodes.

Whether claims 39, 52, and 65 are patentable under 35 U.S.C. §103(a) over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Junkin, U.S. Patent No. 6,493,717 (Junkin) and Harel, U.S. Patent No. 5,873,081 (Harel).

Whether claims 40, 53, and 66 are patentable under 35 U.S.C. §103(a) over Callahan, Epperson, Posse, Bhargava, Junkin, and Harel as applied to claims 39, 52, and 65 respectively, and further in view of Zhou et al., U.S. Patent No. 5,995,511 (Zhou).

Whether claims 41, 42, 54, 55, 67, and 68 are patentable under 35 U.S.C. §103(a) over Callahan, Epperson, Posse, Bhargava, Junkin, Harel, and Zhou as applied to claims 40, 53, and 66, and further in view of Bamford et al., U.S. Patent No. 6,243,702 (Bamford).

Whether claims 44, 57, and 70 are patentable under 35 U.S.C. §1039a) over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Naidu.

Whether claims 46, 48, 59, 61, 72, and 74 are patentable under 35 U.S.C. §103(a) over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Filepp.

Whether claims 47, 60, and 73 are patentable under 35 U.S.C. §103(a) over Callahan, Epperson, Posse, and Bhargava as applied to claims 36, 49, and 62 respectively, and further in view of Carey.

VII. GROUPING OF CLAIMS

The rejected claims do not stand or fall together. Separate arguments are presented for the patentability of the claims in each of the following claim groups:

Group I: Claims 36, 49, 62, 43, 49, 56, 62, 69, and 71;

Group II: Claims 45, 48, and 71;

Group III: Claims 38, 40, 51, and 64;

Group IV: Claims 39, 52, 53, 65, and 66;

Group V: Claims 41, 42, 54, 55, 67, and 68;

Group VI: Claims 44, 57, and 70;

Group VII: Claims 46, 48, 59, 61, 72, and 74; and

Group VII: Claims 47, 60, and 73.

VIII. ARGUMENTS

A. Rejections Based on 35 U.S.C. § 112

The Final Office Action rejects claims 38, 40, 51, 53, 64, and 66 under 35 U.S.C. § 112 as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicants regard as the invention.

According to the Office Action, the term "operator" may designate a person or an abstract operator, and that the language in these claims should be modified to distinguish between these two possible meanings of the word "operator".

Reference is made to a telephonic interview between Examiner Dodds and the Applicants' attorney, Victor G. Cooper on May 27, 2003. The Applicants' attorney acknowledges that he indicated that in this interview, he agreed that the use of the term "logical operator" in the claims instead of the term "operator" would be acceptable. However, on reconsideration, the Applicants' attorney believes that such a change would do more harm than good, and declines to make such amendments.

All of the claims at issue use the term "operator trees", which cannot be reasonably confused with a "person" as suggested. The Applicants' attorney believes it would be more confusing, and arguably less definite to amend all use of the term "operator" in this case. Further, use of the term "logical operator" may be misconstrued by some as a substantive limitation of the claims (perhaps excluding functional operators and the like).

B. Rejections Based on 35 U.S.C. § 103

1. The References

a) The Callahan Reference

U.S. Patent No. 6,230,313, issued May 8, 2001 to Callahan et al. discloses a system for conducting performance analysis for executing tasks. The analysis involves generating a variety of trace information related to performance measures, including parallelism-related information, during execution of the task. In order to generate the trace information, target source code of interest is compiled in such a manner that executing the resulting executable code will generate execution trace information composed of a series of events. Each event stores trace information related to a variety of performance measures for the one or more processors and protection domains used. After the execution trace information has been generated, the system can use that trace information and a trace information description file to produce useful performance measure information. The trace information description file contains information that describes the types of execution events as well as the structure of the stored information. The system uses the trace information description file to organize the information in the trace information file, extracts a variety of types of performance measure information from the organized trace information, and formats the extracted information for display. The system can use default or user-defined functions to extract and format trace information for display. After the system displays one or more types of performance measure information, a user of the system can then interact with the system in a variety of ways to obtain other useful performance analysis information.

b) The Epperson Reference

U.S. Patent No. 5,754,771, issued May 19, 1998 to Epperson et al. discloses a maximum receive capacity specifying query processing client/server system replying up to the capacity and sending the remainder upon subsequent request. An Interactive Television (ITV) Client/Server system comprises one or more Clients (e.g., set top boxes or "STBs") connected to a Server module, via a Broadband Communication Network, is described. The system provides a generic mechanism for deciding how much or how little data is to be sent in response to a request from a Client. Specifically, each Client is allowed to describe ahead of time how much memory (approximately or

exactly) it is allowing for the query result. This is communicated by the Client as part of the query itself. Regardless of what other activity the Client undertakes, the server maintains the context of the original query, until the Client terminates the connection. In a similar manner, when other simultaneous queries are open, the context is maintained for each query until the connection is dropped (or the query terminates). The Client can ask for more data (i.e., rows) on any open query. To optimize operation of the environment, requests from the Clients (i.e., queries) are de-multiplexed down to a set of worker threads, available at the server, which carry out the actual work of query processing. Because of this approach, a request for more data from a Client (i.e., set top box) can "land" on any worker thread. This gives queries the ability to "jump" from one thread of execution to another. In this manner, a finite number of threads may be employed to service a large query pool, with maximum throughput.

c) The Posse Reference

U.S. Patent No. 5,544,175, issued August 6, 1996 to Posse discloses a method and apparatus for the capturing and characterization of high-speed digital information. A digital signal detector for sampling the state of a high speed digital signal occurring at a test node in a digital circuit which exhibits the same behavior with repeated applications of the same inputs. The digital signal detector samples the state of a test circuit node at discrete intervals in time and stores the digital levels with timing reference information. The stored information is then compared to the expected behavior of the tested node for analysis of delay-type and other parametric faults, such as performance faults. The digital signal detector includes a state discriminator which determines the state of the input digital signal by comparing its voltage level to one or more threshold voltages. The digital signal detector also includes a memory unit which stores the signal state information upon receipt of a memory strobe signal generated by a memory delay control circuit which delays the application of the memory strobe signal by a predetermined amount of time after the receipt of a clock pulse. The stored digital signal states can be later retrieved from the memory unit for evaluation of the input digital signal.

d) The Bhargava Reference

U.S. Patent No. 5,680,603, issued October 21, 1997 to Bhargava et al. discloses a method and apparatus for reordering complex SQL queries containing inner and outer join operations. A method and apparatus for reordering complex SQL queries containing joins, outer and full outer joins. The method and apparatus first translates the query into a hypergraph representation. Required sets, conflict sets and preserved sets are then generated for the query hypergraph. Using the required sets, a plurality of plans are enumerated, wherein the plans represent associative reorderings of relations in the query. SQL operators are selectively assigned to each of the enumerated plans using the conflict sets and/or preserved sets, so that the results from the plans are identical to the original query. A novel Modified General Outer Join (MGOJ) operator may be assigned to the root of a sub-tree, wherein the MGOJ operator is a compensation operator. The operator assignment is performed recursively for the root of each sub-tree in the plan. One of the enumerated plans (generally the most optimal) is then selected for execution.

e) The Kimmerly Reference

U.S. Patent No. 5,628,017, issued May 6, 1997 to Kimmerly et al. discloses a method and system for providing event-response capabilities to pseudocode. A method and system for providing event-response and monitoring capabilities to a pseudocode program operating in a message or event-based operating environment. The method and system allow the pseudocode program to asynchronously respond to events via one or more trap routines located in the pseudocode program. The method and system also allow the pseudocode program to synchronously monitor events and receive parameters from operating system routines via callback routines located in the pseudocode program. The system includes an execution engine for executing instructions of the pseudocode program. An event-response routine in the pseudocode program includes instructions for responding to the event. An event-response dispatcher is adapted to receive information identifying the event-response routine, save the execution state of the execution engine, cause the execution engine to execute the event-response routine, and restore the execution state of the execution engine that existed before the event-response routine was executed. A transfer routine accessible in response to a call triggered by the event accesses the event-response dispatcher, passes

information identifying the event-response routine to the event-response dispatcher, and returns control to the execution engine after the event-response dispatcher restores the state of the execution engine and returns to the transfer routine.

f) The Rhodes Reference

U.S. Patent No. 6,073,110, issued June 6, 2000 to Rhodes et al. discloses an activity based equipment scheduling method and system. A computer based equipment scheduling system uses activity definition data to schedule equipment. The method may be carried out by a networked computer system that receives activity definition data representing an activity, such as a volleyball game or board meeting, that occurs in a building. This data is stored in a database for use by multiple nodes. The activity definition data may include text or graphical data indicating an activity name, data representing a list of zones that the activity will affect and data representing the mode of possible operation of each building zone affected by the activity. The method also includes assigning zone mode data to the activity data for use in controlling building resources and then automatically controlling the building resources to attain a zone mode of operation assigned to the activity definition data.

g) The Junkin Reference

U.S. Patent No. 6,493,717, issued December 10, 2002 to Junkin discloses a system and method for managing database information to be presented in HTML format for retrieval and display by a Web browser. Database information is managed by responding to a user selection delivered by browser software by retrieving the contents of portions of a database and constructing an HTML-compatible presentation of the contents in accordance with definitions that hierarchically link the portions separately from any database-provided links between the portions.

h) The Harel Reference

U.S. Patent No. 5,873,081, issued February 16, 1999 to Harel discloses a method and mechanism for filtering incoming documents against user queries. A plurality of user queries including terms connected by logical operators is received. Terms and sub-expressions are combined

into distinct sub-expressions and embedded into a directed acyclic graph (DAG) having a plurality of nodes. Each node in the DAG includes pointers to any successor nodes thereof, the terms in the queries are embedded as source nodes in the graph, and the operators embedded as internal nodes. When a document is received, the document is evaluated against the nodes in the DAG by comparing the relevant terms in the document with the source nodes in the DAG representative thereof. For each term that matches a source node, the internal successor node of the matched source node is evaluated based on the logical operator represented by the successor node and truth information of the predecessor nodes thereto, thereby determining a truth value of the internal successor node. Information is returned indicative of which of the successor nodes were evaluated as true. From that information, the queries which matched the document and the users corresponding thereto can be determined.

i) The Zhou Reference

U.S. Patent No. 5,995,511, issued November 30, 1999 to Zhou et al. discloses a queue control system for use in connection with the transfer of information, in the form of information transfer units, in a digital network. The network provides a plurality of service rate classes, based on, for example transmission rates for the various paths. The information buffer control subsystem includes a information transfer unit receiver, a information transfer unit buffer and a group controller. The information transfer unit receiver receives the information transfer units, and the buffer is provided to buffer the received information transfer units prior to transmission. The group controller controls the buffering of information transfer units received by the information transfer unit receiver in the buffer. In that operation, the group controller aggregates the information transfer units for each path in the buffer according to respective service rate classes, in particular aggregating the information transfer units for each path in a queue and further aggregating the queues for the paths associated with each service rate class in a queue. A transmission scheduler is also disclosed for use in transferring information, in the form of information transfer units, each associated with a path, in a digital network. The network provides a plurality of service rate classes, based on, for example, transmission rates for the various paths. The information transfer units for each path in a path queue, and the path queues for the paths associated with each service rate class

are aggregated in a service rate queue. The transmission scheduler includes a information transfer unit selector for selecting from among the service rate queues, one path queue to provide a information transfer unit for transmission, and a information transfer unit transmitter for transmitting the information transfer unit provided by the selected path queue.

j) The Bamford Reference

U.S. Patent No. 6,243,702, issued June 5, 2001 to Bamford et al. discloses a method and system for removing propagation delays between a plurality of database servers that have access to a common database is provided. According to the method, each database server is associated with a logical clock. In response to initiating a commit of a transaction executing on a database server, a commit time for the transaction is determined and broadcast to one or more other database servers. The broadcast is overlapped with a transaction log force. Upon receiving the commit time, the database servers compare the transmitted commit time to the time indicated by their logical clock. If the commit time is greater than the time indicated by their logical clock, the database server sets its logical time to reflect a time that is at least as recent as the time reflected by the transmitted commit time.

k) The Naidu Reference

U.S. Patent No. 5,752,002, issued May 12, 1998 to Naidu et al. discloses a method and apparatus for optimizing performance of a computer system component design by performance analysis of a simulation of the design. The method of the present invention comprises providing the computer system component design to an analyzing apparatus and carrying out a simulation run of the design. During the simulation run, operation data is generated cycle by cycle, and the generated operation data is collected and stored in a log file. The log file is input to a parser and the operation data is sequentially parsed to produce parsed data. Statistical calculations are then performed on the parsed data, and the performance results are output to the designer in graphical form. The performance information can be used to enhance performance of the computer system component prior to its fabrication.

l) The Filepp Reference

U.S. Patent No. 5,594,910, issued January 14, 1997 to Filepp et al. discloses a distributed processing, interactive computer network and method of operation. The network is designed to provide very large numbers of simultaneous users access to large numbers of applications which feature interactive text/graphic sessions. The network includes one or more host computers having application data stores; a plurality of concentrator computers, also including application data stores, the concentrator computers being connected in groups of one of more to each of the host computers; and a plurality of reception system computers connected in groups of one or more to each of the concentrator computers, the reception system computers being arranged so that respective users can request interactive applications at the reception system computers. In accordance with the design, the reception system computers also include application data stores. The method for operating the network includes steps for generating the interactive text/graphic sessions from objects that include data and/or program instructions. Additionally, the method features steps for distributing objects among the data stores of the network computers, and, thereafter, permitting the reception system computer at which an application is requested to selectively collect objects required for the application from the network and the respective reception system so that the requested application may be presented at the reception system based on the objects collected. This operation decreases processing demand on the higher-level network elements, permitting them to function primarily as data supply and maintenance resources, thereby reducing network complexity, cost and response time.

m) The Carey Reference

U.S. Patent No. 6,285,997, issued September 4, 2001 to Carey et al. discloses a method, apparatus, and article of manufacture for a computer-implemented technique for query optimization with deferred updates and autonomous sources. An object-oriented query is executed to retrieve data from a database. The database is stored on a data storage device connected to a computer. The object-oriented query is transformed into subqueries, wherein at least one subquery is directed against a database, and wherein one subquery is directed against an object cache. Each subquery that is directed against a database is executed to retrieve data from the database into the object cache.

The subquery that is directed against the object cache is executed to retrieve data for the query, wherein the data incorporates updates to the object cache and updates to the database.

2. Claims 36, 43, 45, 49, 56, 58, 62, 69, and 71 are Patentable Over the Cited References

Stated as economically as possible, the Final Office Action's rejection of claim 36 violates a basic principle of patentability:

"[I]t is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious. ... This court has previously stated that '[o]ne cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.'" *In Re Fritch*, 972 F.2d 1260, 23 USPQ2d 1780 (Fed. Cir. 1992).

Claim 36 stands rejected as obvious over four references, Callahan, Epperson, Posse, and Bhargava. Claim 36 recites:

> *A method of monitoring an execution of a query performed by a database system having a query coordinator and at least one data server, wherein the query execution comprises at least one execution thread, the method comprising the steps of:*
> *for each thread, generating first execution trace information in the query coordinator, wherein the first execution trace information comprises an execution plan in terms of one or more operator trees;*
> *for each thread, generating second execution trace information in the data server; and*
> *writing the first execution trace information and the second execution trace information to at least one execution log file. (Emphasis added)*

As pointed out in the Amendment filed April 14, 2003, it has already been acknowledged that the Callahan, Epperson, and Posse references do not teach generating *first execution trace information comprising an execution plan in terms of one or more operating trees.*

Instead, the first rejection, as well as the Final Rejection, rely on the Bhargava reference, which teaches the use of execution plans and operator trees as follows:

> **FIGS. 10A, 10B, and 10C together are a flowchart illustrating the method of performing conflict-free operator assignments for the association trees according to the present invention; and** 25

and;

> present invention. The conflict-free operator assignment attempts to generate an operator tree for a given association
> 55 tree by assigning joins, outer joins and full outer joins to the interior nodes of the association tree. If it is not possible to

The Applicants agree that the foregoing teach the use of execution plans and teaches operator trees. But the foregoing does not suggest generating execution trace information in the form of operator trees. The issues then are:

1. Whether the references, when combined, collectively teach generating *first execution trace information comprising an execution plan in terms of one or more operating trees.*
2. Whether it can be fairly said that there is a teaching or suggestion to combine the reference to arrive at that teaching.

Turning to the first issue, the First Office Action argued that this substitution would be obvious to one of ordinary skill in the art because Callahan uses "tasks" and Bhargava "use[s] nodes."

In response, the Applicants clarified what Callahan teaches and what it does not. In particular, the Applicants pointed out that Callahan might be argued to use "tasks" does not report trace information in the form of "tasks" and could not be combined with Bhargava as the First Office Action suggested:

> To be precise, Callahan generates raw trace information, that is later modified by the trace information display or (TID) for presentation to the user. The raw trace information includes information such as that which is illustrated in FIG. 4. In Callahan, each thread does not report trace information in the form of "tasks", but rather, as *raw* information that is later assembled and formatted into a form is desired by the user.
>
> Callahan uses "tasks" but clearly, does not report trace information in the form of "tasks." Callahan therefore either (1) teaches that there is no relationship between the "task" organization and the form of the reported execution trace information, or (2) teaches that the execution trace information should be reported *differently*. In either case, Callahan can only be said to teach away from claim 1.
>
> Given the foregoing, it does not follow that if the Bhargava, which the Office Action indicates "use[s] nodes", would obviously report trace information the form of execution plans having one or more operator trees.

This issue is readdressed in paragraph 22 of the Final Office Action. Relying on the same portions of the Bhargava reference, the Final Office Action remarks:

> "Again, Bhargava teaches both elements, the use of an application plan, which is essentially the same as an execution plan and the use of an operator tree. It would have been obvious to one of ordinary skill at the time the invention to infer that the operator trees used in independent claims 36, 49, and 62 had nodes associated with the trees." (Final Office Action, page 15, paragraph 22)

The Applicants respectfully disagree. Bhargava may teach the use of an application plan, but that is a long sight away from what is relevant here ... namely, whether Bhargava teaches generating execution trace information which *comprises* an execution plan. Bhargava also teaches the generation of an operator tree, but it does *not* teach generating execution trace information *comprising* an execution plan *in terms of one or more operating trees.*

The Applicants do not understand the meaning of the statement that it would have been obvious to "infer that the operator trees used in independent claims 36, 49, and 62 had nodes associated with the trees." The issue is whether, when combined, the Callahan, Epperson, Posse, and Bhargava reference together teach generating execution trace information comprising an execution plan in terms of one or more operating trees, and where there is any teaching to combine those references.

Referring back to the primary reference (Callahan), the Applicants have pointed out that although the reference uses "tasks" as the First Office Action suggested, it clearly does not report trace information in form of "tasks". Therefore, Callahan teaches either (1) that there is no relationship between organization of tasks and the form of the reported trace information, or (2) that the execution trace information should be reported differently. In either case, Callahan may teach generating execution trace information, but cannot be fairly said to teach generating execution trace information *comprising an execution plan in terms of one or more operating trees.* Bhargava is of no help here, since all it does is disclose the use of operator trees.

Essentially, the Final Office Action, like the First Office Action, rejects claim 36 by picking and choosing phrases from a host of references (in this case, four), without regard for the context of these phrases or how they are used in the claim. However, 35 U.S.C. § 103 requires that (1)

combined, the references much teach the Applicants' invention and (2) that there is a teaching or suggestion in the prior art to make that combination. Simply put, there is no teaching anywhere in the art to generate execution trace information in the form of execution plans, and certainly not execution plans expressed in terms of one or more operating trees. Callahan itself teaches otherwise.

Claim 36 also recites that the first execution trace information is generated in the query coordinator and the second execution trace information is generated in the data server. The Final Office Action argues that this is taught by Epperson because it "teaches the use of a query coordinator and a data server", but the claim itself specifies more than just a "query coordinator" and a "data server".

Instead, the issue is whether Epperson (or any of the other 4 references) teach generating first execution trace information in a query coordinator and second execution trace information in a data server. Even if the Epperson reference were to teach a query coordinator and a data server, that alone is not a sufficient basis to reject claim 36.

Also, with respect to claim 36, the Posse reference plainly does not teach writing execution trace records to an execution log file:

> At each successive execution of the test program, the value of the low-order address bits is incremented by one. This advances the sampling of the digital signal (which is regenerated with each test program execution) in discrete intervals. Referring to the above example, during the next

and,

> In a preferred embodiment of the present invention, tester 500 interfaces with a simulator 508. Simulator 508 provides reference data representing the expected behavior of test node 512 to control unit 502 via reference data line 507. Typically, the simulation of digital circuit 515 is performed prior to (and independently of) the testing of digital circuit 513. The results of the simulation are then provided to control unit 502 in what is referred to as a simulation log file. The simulation log file contains threshold crossings and timing reference data for the test node 512 based upon the application of a given set of test inputs 513 to digital circuit 515. In the preferred embodiment of the present invention,

The foregoing refers to a simulator 508 which interfaces with a circuit tester 500. Results from the simulation are provided to a control unit 502 in what is referred to as a simulation log file,

which has threshold crossings and timing reference data for test nodes based upon the application of a given set of test inputs to a digital circuit.

The Final Office Action suggests that this reference is used to show the use of an execution log file, but the "simulation log file" of the Posse reference is hardly analogous. The Applicants respectfully suggest that this is yet another example of picking phrases from widely different references in an exercise of hindsight reconstruction.

The Posse reference refers to an entirely different art than that of the Applicants' invention, and refers to collecting *digital circuit simulation* results, not execution trace information with operator trees. The Applicants respectfully disagree that one of ordinary skill in the art would be motivated to alter the teachings off Callahan, Epperson, and Bhargava as described in an unrelated reference in an entirely different art (digital circuit simulation and analysis).

Claims 49 and 62 recite limitations analogous to those of claim 36 and are patentable on the same basis.

Claims 43, 49, 56, 62, 69, and 71 include limitations analogous to those of claim 36, and are patentable for the same reasons.

3. Claims 45, 48, and 71 are Patentable Over the Cited References

According to the Office Action, the step of reconstructing execution trace information from a log file is disclosed in the Posse reference. However,

- column 8, lines 28-33 of the Posse reference refers to reconstructing a test period of an input signal;
- column 9, lines 27-29 of the Posse reference refers to testing of a digital signal;
- column 10, lines 1-3 of the Posse reference describes engineering models that generate library files containing threshold crossing and "dependency" and "condition information"; and
- column 10, lines 54-57 of the Posse reference disclose reconstructing simulation data.

The Final Office Action alleges that this teaches reconstructing test periods using trace information, which is stored in log files, which is "very close" to the technology in claims 45, 58, and 71.

The Applicants respectfully disagree. First, the Posse reference does not describe an art that is analogous to that of the Applicants' claims or the other cited references. Second, the Posse reference discusses a *simulation* log file, which is hardly "close" to that of an execution trace file. Finally, nothing in this reference discloses reconstructing execution trace information with operator trees from an execution log file, as the claims require.

4. Claims 38, 51, and 64 are Patentable Over the Cited References

Claim 38 recites that the execution trace information further comprises operator dispatch information.

The First Office Action suggests that Kimmerly discloses the use of operator dispatch information at column 3, lines 9-16. However, Kimmerly does not disclose including operator dispatch information in execution trace information, as claim 38 recites.

In paragraph 28, the Final Office Action answers this by indicating that Kimmerly teaches the use of operator dispatch information and that "Epperson teaches the use of Execution trace information."

Again, this is a matter of using hindsight to choose phrases from disparate reference and simply combining them together to reject the claims. Claim 38 recites that the execution trace information *comprises* operator dispatch information, operator start times, and operator stop times. None of the references disclose this feature.

As for the motivation to combine the references, the Final Office Action suggests that this can be found because "Callahan, Epperson, Posse, Bhargava, and Kimmerly teach the execution of tasks and Callahan, Epperson, and Kimmerly teach the use of routines and subroutines." The Applicants are not sure how this provides the motivation to combine/modify the references as suggested.

Claims 51 and 64 include limitations analogous to those of claim 38, and are patentable for the same reasons.

5. Claims 39, 40, 52, 53, 65, and 66 are Allowable

Claim 39 recites that the execution trace information includes a session identifier and a query ID, while claim 40 recites that the execution trace record includes an operator ID, a start timestamp, and a finish timestamp . Here again, the Office Action rejects these claims based upon references that at best, merely discloses a parameter, but does not disclose providing that parameter in anything analogous to an execution trace record.

As for a rationale to combine these references, the Office Action indicates:

> "It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Jurkin and with Callahan, Epperson, and Posse since Callahan, Epperson, Posse, and Jurkin teach the execution of tasks in the use of trace information, Callahan, Epperson, and Jurkin teach the use of networks and the use of routines and subroutines, Epperson and jerkin teach the performing of query is, the use of databases, and the use of servers, posse and Jurkin teach the use of nodes."

The Applicants respectfully disagree that the foregoing rationale for combine these references comports with the requirements of 35 U.S.C. § 103. Even if the above statements were true in terms of what each reference teaches, it does not provide a rationale for combine or modify these references. Accordingly, the Applicants respectfully traverse the rejection of claims 39 and 40.

Claims 52 and 65 recite features analogous to those of claim 39, and are patentable on the same basis.

Claims 53 and 66 recite features analogous to those of claim 40, and are patentable on the same basis.

6. Claims 41, 42, 54, 55, 67, and 68 are Allowable

According to the Office Action, the Zhou reference teaches the start timestamps and finish timestamps. However, again, those timestamps are not used in anything analogous to an execution trace record, and no motivation to include those timestamps in execution trace record is provided.

The Final Office Action suggests that the references may be combined because they "share many elements in common", but merely sharing common elements is an insufficient motivation to combine references.

Hence, the Applicants respectfully traverse these rejections.

7. Claims 44, 57, and 70 are Allowable

According to the Office Action, Naidu teaches writing to different files at different times. Again, this teaching is out of context with anything analogous to an execution trace record. Further, the suggested motivation to combine the references:

"Callahan, Epperson, Posse, Bhargava, and Naidu teach the execution of tasks and the use of testing, Callahan, Epperson, Posse, and Naidu teach the use of time as an element, Epperson, Posse, Bhargava, and Naidu use performance analysis, and Callahan, Epperson, and Naidu teach the use of routines".

8. Claims 46, 48, 59, 61, 72, and 74 are Allowable

The Office Action indicates that Filepp teaches the use of presentation commands and timestamps, but, as before, not in the context of assembling execution trace information. Further, no mention is made of synchronizing execution trace record as according to the timestamp, as recited in claim 48.

The Final Office Action fails to provide a motivation to combine the references, relying instead on a simple statement of what they disclose. A rejection under 35 U.S.C. § 103 requires more than simply finding features in prior art references ... it requires a rationale indicating why one of ordinary skill in the art would have been motivated to combine or modify the references as indicated.

Accordingly, the Applicants respectfully traverse these rejections.

9. Claims 47, 60, and 73 are Allowable

The Office Action indicates that the Carey reference teaches performing other routines while executing a query. However, the subject claims recite that execution trace information are generated in two separate entities (the query coordinator and the data server) while executing the query. Even if the Carey reference discloses performing other routines while executing a query, it does not disclose all of the features of the subject claims. Accordingly, the Applicants respectfully traverse these rejections.

IX. CONCLUSION

In light of the above arguments, Appellants respectfully submit that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP

Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: November 3, 2003

By: _____
Name: Victor G. Cooper
Reg. No.: 39,641

VGC/io

G&C 30145.344-US-01

APPENDIX

36. (PREVIOUSLY AMENDED) A method of monitoring an execution of a query performed by a database system having a query coordinator and at least one data server, wherein the query execution comprises at least one execution thread, the method comprising the steps of:

for each thread, generating first execution trace information in the query coordinator, wherein the first execution trace information comprises an execution plan in terms of one or more operator trees;

for each thread, generating second execution trace information in the data server; and

writing the first execution trace information and the second execution trace information to at least one execution log file.

37. (CANCELED)

38. The method of claim 37, wherein the first execution trace information further comprises operator dispatch information, operator start times and operator stop times.

39. The method of claim 36, wherein the second execution trace information includes a session identifier (ID) and a query ID.

40. The method of claim 39, wherein the second execution trace information further includes, for each operator:

an identifier (ID) for the operator;

a start time stamp; and

a finish time stamp.

41. The method of claim 40, wherein the start time stamp and the finish time stamp reference a logical time.

42. The method of claim 40, wherein the start time stamp and the finish time stamp reference a clock time.

43. The method of claim 36, wherein the first execution trace information and the second execution trace information are written to a single file.

44. The method of claim 36, wherein the first execution trace information and the second trace information are written to different files.

45. The method of claim 36, further comprising the step of reconstructing the execution trace information from the log file.

46. The method of claim 36, further comprising the steps of:

accepting a presentation command;

reconstructing the execution trace information according to the thread ID, the time stamp and the presentation command; and

presenting the reconstructed execution trace information.

47. The method of claim 36, wherein the steps of generating first execution trace information in the query coordinator and generating second execution trace information in the data server is performed while executing the query.

48. The method of claim 36, wherein the first execution trace information and the second execution trace information include a thread ID and a time stamp, and the method further comprises the step of:

synchronizing the execution trace records according to the time stamp.

49. An apparatus for monitoring an execution of a query performed by a database system having a query coordinator and at least one data server, wherein the query execution comprises at least one execution thread, the apparatus comprising:

means for generating first execution trace information for each thread in the query coordinator, wherein the first execution trace information comprises an execution plan in terms of one or more operator trees;

means for generating second execution trace information for each thread in the data server; and

means for writing the first execution trace information and the second execution trace information to at least one execution log file.

50. (CANCELED)

51. The apparatus of claim 50, wherein the first execution trace information further comprises operator dispatch information, operator start times and operator stop times.

52. The apparatus of claim 49, wherein the second execution trace information includes a session identifier (ID) and a query ID.

53. The apparatus of claim 52, wherein the second execution trace information further includes, for each operator:

an identifier (ID) for the operator;

a start time stamp; and

a finish time stamp.

54. The apparatus of claim 53, wherein the start time stamp and the finish time stamp reference a logical time.

55. The apparatus of claim 53, wherein the start time stamp and the finish time stamp reference a clock time.

56. The apparatus of claim 49, wherein the first execution trace information and the second execution trace information are written to a single file.

57. The apparatus of claim 49, wherein the first execution trace information and the second trace information are written to different files.

58. The apparatus of claim 49, further comprising means for reconstructing the execution trace information from the log file.

59. The apparatus of claim 49, further comprising:

means for accepting a presentation command;

means for reconstructing the execution trace information according to the thread ID, the time stamp and the presentation command; and

means for presenting the reconstructed execution trace information.

60. The apparatus of claim 49, wherein first execution trace information is generated in the query coordinator and the second execution trace information is generated in the data server while the query is executed.

61. The apparatus of claim 49, wherein the first execution trace information and the second execution trace information include a thread ID and a time stamp, and the apparatus further comprises:

means for synchronizing the execution trace records according to the time stamp.

62. (PREVIOUSLY AMENDED) A program storage device, readable by a computer, tangibly embodying at least one program of instructions executable by the computer to perform method steps of monitoring an execution of a query performed by a database system having a query coordinator and at least one data server, wherein the query execution comprises at least one execution thread, the method steps comprising the steps of:

for each thread, generating first execution trace information in the query coordinator, wherein the first execution trace information comprises an execution plan in terms of one or more operator trees;

for each thread, generating second execution trace information in the data server; and

writing the first execution trace information and the second execution trace information to at least one execution log file.

63. (CANCELED)

64. The program storage device of claim 63, wherein the first execution trace information further comprises operator dispatch information, operator start times and operator stop times.

65. The program storage device of claim 62, wherein the second execution trace information includes a session identifier (ID) and a query ID.

66. The program storage device of claim 65, wherein the second execution trace information further includes, for each operator:

an identifier (ID) for the operator;

a start time stamp; and

a finish time stamp.

67. The program storage device of claim 66, wherein the start time stamp and the finish time stamp reference a logical time.

68. The program storage device of claim 66, wherein the start time stamp and the finish time stamp reference a clock time.

69. The program storage device of claim 62, wherein the first execution trace information and the second execution trace information are written to a single file.

70. The program storage device of claim 62, wherein the first execution trace information and the second trace information are written to different files.

71. The program storage device of claim 62, further comprising the step of reconstructing the execution trace information from the log file.

72. The program storage device of claim 62, wherein the method steps further comprise the steps of:

accepting a presentation command;

reconstructing the execution trace information according to the thread ID, the time stamp and the presentation command; and

presenting the reconstructed execution trace information.

73. The program storage device of claim 62, wherein the method steps of generating first execution trace information in the query coordinator and generating second execution trace information in the data server is performed while executing the query.

74. The program storage device of claim 62, wherein the first execution trace information and the second execution trace information include a thread ID and a time stamp, and the method steps further comprise the step of:

synchronizing the execution trace records according to the time stamp.